Significance of network metrics

Ramon Ferrer-i-Cancho & Argimiro Arratia

Universitat Politècnica de Catalunya

Version 0.4
Complex and Social Networks (2020-2021)
Master in Innovation and Research in Informatics (MIRI)

Official website: `www.cs.upc.edu/~csn/`
Contact:

- Ramon Ferrer-i-Cancho, rferrericancho@cs.upc.edu, `http://www.cs.upc.edu/~rferrericancho/`
- Argimiro Arratia, argimiro@cs.upc.edu, `http://www.cs.upc.edu/~argimiro/`

Hypothesis testing

Monte Carlo methods

Generation of random graphs

Qualitative hypothesis testing

Some rules:

- Clustering is significantly high if $C \gg C_{ER}$.
- Distance is small (small-world phenomenon) if $l \approx logN$.

But

- Clustering might be significantly high even if $C \gg C_{ER}$ does not hold.
- In small networks, numerical differences between the true values and those of the null hypothesis are smaller. Comparison of numbers no longer works.

Goal: turning the reasoning more rigorous.

Hypothesis testing I

- x: network metric (e.g., clustering coefficient, degree correlation, ...).
- Is the value of x significant? (with regard to what?)
- Is the value of x significant with regard to a certain null hypothesis? But which one?
- Three kinds of questions:
    - Is x significantly low? e.g., is the mean minimum vertex-vertex distance significantly low? ("small-wordness").
    - Is x significantly high? e.g., is the clustering coefficient significantly high?
    - Is $|x|$ significantly high? e.g., is the degree correlation strong enough?

Families of null hypotheses

Random pairing of vertices chosen uniformly at random (Erdös-Rényi graph).

- Variable number of edges (parameters N and π). The $G(N, \pi)$ model.
- Constant number of edges (parameters N and M, the number of edges). The $G(N, M)$ model.

Problem: unrealistic degree distribution!

Random pairing of vertices constraining the degree distribution [Newman, 2010]

- A given degree distribution: $p(k_1), p(k_2), ..., p(k_{N_{max}})$ (not seen in this course; similar to $G(N, \pi)$).
- A given degree sequence: $k_1, k_2, ..., k_{N_{max}}$ (similar to $G(N, M)$). The **configuration model** and the **switching model**.

Restating the questions in terms of probabilities

- x_{NH}: value of x in a network under the null hypothesis.
- $p(x_{NH} \leq x)$, $p(x_{NH} \geq x)$ (cumulative probability, distribution functions).
- α: significance level. Typically $\alpha = 0.05$.

Three kinds of questions:

- Is x significantly low? Yes if $p(x_{NH} \leq x) \leq \alpha$.
- Is x significantly high? Yes if $p(x_{NH} \geq x) \leq \alpha$.
- Is $|x|$ significantly high? Yes if $p(|x_{NH}| \geq |x|) \leq \alpha$.

Restating the questions in terms of probabilities

Two approaches:

- Analytical:
    - Calculate $p(x_{NH} \leq x)$, $p(x_{NH} \geq x)$ or $p(|x_{NH}| \geq |x|)$.
    - Problem: it can be mathematically hard specially if one wants to obtain exact results.
- Numerical:
    - Monte Carlo procedure to estimate $p(x_{NH} \leq x)$, $p(x_{NH} \geq x)$ or $p(|x_{NH}| \geq |x|)$.
    - Problem: computationally expensive.

Monte Carlo procedure: example on $p(x_{NH} \geq x)$

$f(x_{NH} \geq x)$: number of times that $x_{NH} \geq x$.
Algorithm with parameters x and T:

1. $f(x_{NH} \geq x) \leftarrow 0$.
2. Repeat T times:
   - Produce a random network following the null hypothesis.
   - Calculate x_{NH} on that network.
   - If $x_{NH} \geq x$ then $f(x_{NH} \geq x) \leftarrow f(x_{NH} \geq x) + 1$.
3. Estimate $p(x_{NH} \geq x)$ as $f(x_{NH} \geq x)/T$.

T must be large enough! $1/T \ll \alpha$

Monte Carlo methods I: uniform random number generators

There are standard algorithms for producing

- Uniformly random natural numbers between 0 and X_{max}.
  - In C, the the function `random()` produces random numbers between 0 and `RAND_MAX`.
- Uniformly (pseudo-real numbers between 0 and 1 (constant p.d.f. between 0 and 1).
  - In C, `random()/double(RAND_MAX)` (better procedures are known).

Monte Carlo methods II: elementary operations for constructing random networks

Choosing a random vertex (assume that vertices are labeled with natural numbers).

- Produce $x \sim U[0, X_{max}]$ (e.g., $X_{max} =$ RAND_MAX).
- Output $x \mod N$ (e.g., `random()%`N)

Problem: innacurate if $X_{max} \mod N \neq 0$.
Alternative: Produce $x \sim U(0, 1)$ and Output xN

Deciding if a pair of vertices are linked.

- Produce $x \sim U[0, 1]$.
- Link the pair iff $x \leq \pi$.

Monte Carlo methods III: generating a uniformly random permutation

- Given a sequence of length n, there are $n!$ possible permutations.
- An algorithm that produces a random permutation that has probability $1/n!$.
- A C++ example: random_shuffle(...)

An algorithm for generating a uniformly random permutation

An algorithm that takes a sequence $x_1, x_2, ..., x_n$ that is updated making that the last $n - m$ last elements are a suffix of the permutation of the sequence of increasing length.

```
1. $m \leftarrow n$
2. Repeat while $m \geq 2$
   2.1 Produce $i$ a uniformly random number between 1 and $m$.
   2.2 Swap $x_i$ and $x_m$.
   2.3 $m \leftarrow m - 1$
```

- Prove that the random permutations are equally likely.
- Important to understand the configuration model.

Erdös-Rényi graph with variable number of edges I

- Naive algorithm: for every pair of nodes u, v, add a link between u and v with probability π (generating a random uniform number between 0 and 1 for every pair).
- Problem: time of the order of N^2
- Possible solution:
  - Generate a degree sequence using a generator of binomial deviates (with N and π as parameters).
  - Produce a random graph using the *configuration model* or a better algorithm.

Problem: the degree sequence must be **graphical**.

Erdös-Rényi graph with variable number of edges II

A degree sequence $k_1, k_2, ..., k_i, ..., k_N$, with

- $k_1 \geq k_2 \geq .... \geq k_i \geq ... \geq k_N$
- $0 \leq k_i \leq N - 1$

is graphical (Erdös and Gallai) if and only if

- $$\sum_{i=1}^{N} k_i$$
  is even.
- For every integer r, $1 \leq r \leq N - 1$,
  $$\sum_{i=1}^{r} k_i \leq r(r-1) + \sum_{i=r+1}^{N} min(r, k_i)$$

No need to worry if the degree sequence comes from a real graph.

Be careful with sequences of random numbers!

Erdös-Rényi graph with variable number of edges III

Better algorithm:

- Generate M using a generator of binomial deviates (with $\binom{N}{2}$ and π as parameters, assuming no loops).
- Produce a random graph using an algorithm for generating an Erdös-Rényi graph with constant number of edges (see next).

Erdös-Rényi graph with constant number of edges

- Naive algorithm: choose M pairs of edges. To choose a pair:
  1. Generate a pair of random uniform number between 1 and N.
  2. Choose the pair if the pair has not been chosen before and it is well-formed according to given constraints (on loops, multiple edges...).
- Challenge: checking that the pair has not been chosen before (time and memory cost).

The configuration (or matching) model I

- Input: a degree sequence $k_1, ..., k_i, ..., k_N$
- "stubs: half edges"
- The i-th vertex produces k_i stubs.
- $m = \sum_{i=1}^{N} k_i$ stubs.
- Repeat till there are not available stubs:
  - Choose a pair of stubs x, y uniformly at random.
  - Add a link between x and y.
  - Remove the stubs x and y.
- Implementation: same tricks as algorithm for generating random permutations.
- Example: linear tree of 4 nodes.

The configuration (or matching) model II

Properties:

- Number of pairings that can be formed with m stubs: ? (harder question if we focus on different pairings).
- All possible pairings of "stubs" are equally likely (uniformity as in the algorithm for producing random permutations).
- The networks than can be generated are not necessarily equally likely [Newman, 2010]

The configuration (or matching) model III

How to deal with loops

- An even number of "stubs" is needed (a stub cannot be left unmatched).
- $m = \sum_{i=1}^{N} k_i$ is even if there are loops.
- The handshaking lemma: $\sum_{i=1}^{N} k_i = 2E$.
- Example of network with odd m: two edges $u - v$, $v - v$.
  - u has degree 1 and contributes with one stub.
  - The degree of v is 2? (recall an adjacency matrix definition of vertex degree, $k_i = \sum_{i=1}^{N} a_{ij}$)
  - v should contribute with 3 stubs, not two.
- Adopt the convention that a loop contributes with two to the degree of the node involved [Blitzstein and Diaconis, 2010].
- Loops have two stubs too!

The configuration (or matching) model IV

If the edge is badly-formed according to given constraints (on loops, multiple edges,...):

- Reject the configuration and restart to preserve uniform distribution of matching configurations. Problem: inefficient! (badly formed edges are likely if the degree distribution is **heavy-tailed**, e.g., self-loops involving **hubs** or multiple edges involving hubs are expected).
- Do not restart: choose another random pair of stubs. Problem:
  - Biased sampling (loss of uniformity by increasing the configurations (pairings) with a given prefix or suffix).
  - Backtracking (e.g., linear tree of 4 vertices).

The switching model I

Algorithm

- Input: a network of E edges and Q (a parameter)
- Repeat QE times:
  - Choose two edges uniformly at random: $u - v$ and $s - t$.
  - Exchange the ends to give $u - t$ and $s - v$ if they are well-formed according to given constraints (on loops, multiple edges,...).
  - Failures must be counted for detailed balance. [Milo et al., 2003].

The switching model II

- Easy to adapt to directed networks: exchange the ends of $u \rightarrow v$ and $s \rightarrow t$ to give $u \rightarrow t$ and $s \rightarrow v$ if they are well-formed according to given constraints (on loops, multiple edges,...).
- Fundamental property: the switching preserves degrees (or in-degree and out in-degrees).
- Challenges:
    - The value of Q.
        - Clue: coupon collector's problem.
        - Solution: $Q \sim \log E$ (at least; to warrant that each edge in the original network is chosen at least once).
    - When a switching is not feasible, try another and continue or restart?

The configuration and the switching model

Trade-offs between computational efficiency, statistical properties and complexity of the algorithm:

- Configuration model: uniformity over pairings (not graphs) and computationally expensive (or not usable) due to rejection [Blitzstein and Diaconis, 2010].
- Switching model: usable, but still computationally expensive and uniform sampling is not warranted.
- The generation of random graphs with a given degree sequence is a living field of research [Coolen et al., 2009, Blitzstein and Diaconis, 2010, Roberts and Coolen, 2012].

The switching algorithm with uniform sampling I

The switching algorithm produces a new network a from a network a', preserving the degree distribution.

The original switching algorithm accepts all swaps where valid edges are formed.

To sample uniformly in an undirected graph, the acceptance probability has to be [Coolen et al., 2009]

$$p_{accept}(a|a') = \frac{n(a')}{n(a') + n(a)}$$

where $n(a)$ is the graph mobility, i.e. the number of moves that can be executed on a.

The switching algorithm with uniform sampling II

$$n(a) = \frac{1}{4}K_1(K_1 - 1) - \frac{1}{2}K_2 - \frac{1}{2}\sum_{ij} k_i a_{ij} k_j + \frac{1}{4}Tr(a^4) + \frac{1}{2}Tr(a^3)$$

with

$K_x = \sum_i k_i^x$
$Tr(a) = sum_{i=1}^N a_{ii}$
$(a^k)_{ij}$, the number of walks of length k (base case $k = 1$).

Efficient implementation of the calculation of $n(a)$: $O(N)$ time.

The switching algorithm with uniform sampling III

Protocol:

- Choose four different vertices from a'
- Check whether they form exactly two edges
- Switch the vertices to produce a.
- Accept with probability $p_{accept}(a|a')$.

Further issues:

- Similar methods for directed networks [Roberts and Coolen, 2012]
- Why uniform sampling? Alternatives.

Blitzstein, J. and Diaconis, P. (2010).
A sequential importance sampling algorithm for generating random graphs with prescribed degrees.
Internet Mathematics, 6(4):489–522.

Coolen, A. C. C., De Martino, A., and Annibale, A. (2009).
Constrained Markovian dynamics of random graphs.
Journal of Statistical Physics, 136(6):1035–1067.

Milo, R., Kashtan, N., Itzkovitz, S., Newman, M., and Alon, U. (2003).
On the uniform generation of random graphs with prescribed degree sequences.
arXiv preprint cond-mat/0312028.

Newman, M. E. J. (2010).
Networks. An introduction.
Oxford University Press, Oxford.

Roberts, E. and Coolen, A. (2012).
Unbiased degree-preserving randomization of directed binary networks.
Physical Review E 85, 85:046103.